

Question (1)

- It is required to develop an applet class called **Star** in Java to draw a regular Star (having variable number of equal spikes) in the center of a Frame of size 400 x 350.
- (A) [5 Marks] The class will have an instance variable **spikes**, for number of spikes of the Star. Write method **init** to get the number of spikes from the **param** of the HTML file and assign it in instance variable **spikes** to be used by the **paint** method to draw Star. The value of **spikes** should be between 5 and 10. If it is not in this range, then use 5 as the default value and output suitable message.
- (B) [15 Marks] Write the **paint** method to draw the Star having spikes as the number of spikes. The Star is to be filled with **red color**. Use trigonometry (**sin** and **cos** methods of class **Math**) and **for** loop to calculate the coordinate points of the Star. If the window is resized, the geometric shape should adjust accordingly.
- (C) [5 Marks] Write HTML file for the applet. This file contains **title** as **Red Star**, **param** name = "spikes" with value = "5".

Question (2)

It is required to develop an application in Java for counting number of even and odd integers in a big file using multithreading. Read the file contents in an **ArrayList** (built-in JCF class) object **list**. Elements of **list** are of type **Integer**. Divide the **list** into 3 equal parts. Count even and odd integers in each part of the **list** in individual thread.

- (A) [5 Marks] Write a class called **Count** for counting even integers and odd integers, having two instance variables: **evenCount** and **oddCount** of type **int**. Initialize both instance variables to 0 in the constructor. The class has following additional methods:

setEven:	to set the instance variable evenCount by parameter even .
setOdd:	to set the instance variable oddCount by parameter odd .
getEven:	to return instance variable evenCount .
getOdd:	to return instance variable oddCount .

- (B) [10 Marks] Write a class called **EvenOddCount** that implements interface **Runnable** having instance variables:
- | | |
|--------------------|---|
| startIndex: | starting index of the part of the list to be processed |
| endIndex: | last index of the part of the list to be processed |
| count: | whose elements are to be processed |
| count2: | shared object of type Count to hold the counts of even and odd integers in the list. |

It has a constructor to initialize the instance variables and method **run** that counts the number of even and odd integers in its part of the list. First count in local variables and then updates the shared object **count**. As **count** is a shared object, thread synchronization is required.

- (C) Write a class called **CountEvenOddByMultithreading** having following methods:

- (i) [5 Marks] **inputDataFromFile:** to input the data from a text file and save it in the **list**. The method accepts the name of the data file (**String**) as the first parameter and **list** as the second parameter. The elements of the list are of type **Integer**. Use class **Scanner** to input the data (of type **int**) from the text file. Note that you have to open the data file, create suitable file related objects before reading any data from the data file and you have to close the data file at the end. The method returns the number of data items saved in the **list**. Your method should handle following types of exceptions that may be thrown: **FileNotFoundException**, **NoSuchElementException** and **IllegalStateException**.

- (ii) [12 Marks] **main method:** In the main method, create a list of type **ArrayList**, initial size = 1000. Each element of **ArrayList** object **list** is of type **Integer**. Create a shared object **count** of type **Count**. Input the file name from the user. Call method **inputDataFromFile** to read the data from the file and save it in the **list**. Create suitable threads and call appropriate methods for counting the number of even and odd integers in the **list**. Output the value of these counts.

Use **ExecutorService** class to create a pool of threads in the main. Wait for the termination of all the threads before printing number of even and odd integers.